DEPP 420-56 ASSIGNMENT 1

Primary keys shown.

Foreign keys = FK coded.

Cardinalities in **bold** as **one** to **one** or **one** too **many** in relationships.

Below are:

ERD entities with attributes

Assumptions

Dependencies —>

Relations with PK underlined and coded with FK=.

Relationships with **cardinalities in bold**

CUSTOMER entity

CUS_CODE = PK

CUS_LNAME

CUS_AREACODE

CUS_INITIAL

CUS_BALANCE

CUS_PHONE

CUS_FNAME

Assume:

1. Customer name has only one balance.
2. Enforce one phone per customer - see notes**
3. Not shown here: CUS_BALANCE derived from LINE_PRICE*LINE_UNITS*invoices.

Dependencies:

ERD shows: CUS_CODE —> CUS_FNAME, CUS_INITIAL, CUS_LNAME, CUS_AREACODE, CUS_PHONE, CUS_BALANCE

transitive: CUS_FNAME, CUS_INITIAL, CUS_LNAME —> CUS_AREACODE, CUS_PHONE

transitive: CUS_FNAME, CUS_INITIAL, CUS_LNAME —> CUS_BALANCE

Relations:

CUSTOMER_NAMES(CUS_CODE, CUS_FNAME, CUS_INITIAL, CUS_LNAME)

FK = CUS_FNAME, CUS_INITIAL, CUS_LNAME

CUSTOMER_PHONE(CUS_FNAME, CUS_INITIAL, CUS_LNAME,CUS_AREACODE, CUS_PHONE)

CUSTOMER_BALANCE(CUS_FNAME, CUS_INITIAL, CUS_LNAME, CUS_BALANCE)

Relationships:

CUSTOMER_NAMES can have **one** CUSTOMER_PHONE

CUSTOMER_PHONE assigned to **many** CUSTOMER_NAMES

CUSTOMER_NAMES can have **one** CUSTOMER_BALANCE

CUSTOMER_BALANCE assigned **one** CUSTOMER_NAME

CUSTOMER_NAMES can have **many** INVOICE (see next entity's relations)

INVOICE must have **one** CUSTOMER_NAMES

DEPP 420-56 ASSIGNMENT 1

INVOICE entity
INV_NUMBER = PK
INV_DATE
CUS_DATE

Assume:

1. CUS_DATE is date customer ordered
2. INV_DATE is date customer order is sourced

Dependencies:
INV_NUMBER —> INV_DATE, CUS_DATE

Relation:
INVOICE(INV_NUMBER, INV_DATE, CUS_DATE)

Relationships:
INVOICE can have **many** LINE (see next entity's relations)
LINE must have **one** INVOICE

LINE entity
INV_NUMBER = PK = invoice number
LINE_NUMBER = PK = A line is a line on an invoice.
LINE_UNITS = how many units ordered by customer
P_CODE = product and price code
LINE_PRICE = price per unit, not total for the line.

Assume:
No partial dependency: P_CODE fully functionally dependent on INV_NUMBER, LINE_NUMBER
LINE_PRICE = P_PRICE - P_DISCOUNT derived based on P_CODE on invoice.

Dependencies:
INV_NUMBER, LINE_NUMBER —> LINE_UNITS, P_CODE, LINE_PRICE
transitive: P_CODE —> LINE_PRICE

Relations:
INVOICE_LINE(INV_NUMBER,LINE_NUMBER, LINE_UNITS, P_CODE)
FK = P_CODE
PRICE_PER(P_CODE,LINE_PRICE)

Relationships:
INVOICE_LINE has **one** PRICE_PER
PRICE_PER can have **many** INVOICE_LINE
PRICE_PER has **one** PRODUCT_AT_THIS_PRICE (see next entity's relations)
PRODUCT_AT_THIS_PRICE can have **many** PRICE_PER

DEPP 420-56 ASSIGNMENT 1

PRODUCT entity
P_CODE = PK = multidimensional product price code to determine LINE_PRICE
P_PRICE = price before discount
P_DISCOUNT = discount offered
P_DESCRIPTION = Specific grade; P_DISCOUNT less if P_DESCRIPTION better grade
V_CODE = vendor A charges a different price and offer a different grade
P_QOH = quantity on hand; P_DISCOUNT less if P_QOH less
P_INDATE = next back order date; later means bigger P_DISCOUNT; not a f(P_QOH)
P_MIN = minimum order size from our warehouse; no impact on PRICE or DISCOUNT

Assume:
Customer specifies product and description (grade) in a PCODE that determine discount and via product description determine base price and vendor code (I considered product description determine both base price and product description simultaneously).
P_DESCRIPTION, P_DISCOUNT=f(P_CODE)
P_PRICE = g(P_DESCRIPTION)
V_CODE = f(P_DESCRIPTION)
P_DISCOUNT=g(V_CODE, P_QOH, P_INDATE)

Dependencies:
P_CODE —> P_DESCRIPTION, P_PRICE, V_CODE, P_QOH, P_INDATE, P_DISCOUNT, P_MIN
transitive: P_DESCRIPTION —> P_PRICE
transitive: P_DESCRIPTION —> V_CODE (in DISCOUNT relation)
transitive: V_CODE, P_QOH, P_INDATE —> P_DISCOUNT

Relations:
PRODUCT_AT_THIS_PRICE(P_CODE, P_DESCRIPTION, P_QOH, P_INDATE)
FK = P_DESCRIPTION
FK composite = P_DESCRIPTION, P_QOH, P_INDATE
BASE_PRICE(P_DESCRIPTION, P_PRICE)
DISCOUNT(P_DESCRIPTION, P_QOH, P_INDATE, PDISCOUNT)
FK = P_DESCRIPTION
VENDORS(P_DESCRIPTION, V_CODE)

Relationships:
PRODUCT_AT_THIS_PRICE is a factor of **one** BASE_PRICE
BASE_PRICE is a factor yielding **many** PRODUCT_AT_THIS_PRICE
PRODUCT_AT_THIS_PRICE is a factor of **one** DISCOUNT
DISCOUNT is a factor yielding **many** PRODUCT_AT_THIS_PRICE
DISCOUNT can be achieved from **many** VENDORS
VENDORS offer **one** DISCOUNT
PRODUCT_AT_THIS_PRICE sources from **many** VENDOR_REQUEST (see next entity's relations)
VENDOR_REQUEST are for **one** PRODUCT_AT_THIS_PRICE

DEPP 420-56 ASSIGNMENT 1

VENDOR entity
V_CODE
V_ORDER
V_NAME
V_STATE
V_CONTACT
V_AREACODE
V_PHONE

Assume:

1. V_CODE is unique combination of V_NAME vendor, V_ORDER type
2. V_ORDER is code for order details not documented here: product, speed of delivery, etc
3. Each V_NAME only has one V_CONTACT and V_STATE
4. Each V_CONTACT has one V_AREACODE and V_PHONE
5. A contact can help us order more than once.

Dependencies:
V_CODE —> V_ORDER, V_NAME, V_STATE, V_CONTACT, V_AREACODE, V_PHONE,,
transitive: V_NAME —> V_STATE, V_CONTACT
transitive: V_CONTACT —> V_AREACODE, V_PHONE

Relations:
VENDOR_REQUEST(V_CODE, V_ORDER, V_NAME)
FK = V_NAME
WAREHOUSE(VNAME, V_STATE, V_CONTACT)
FK = V_CONTACT
SALES_PERSON(V_CONTACT, V_AREACODE, V_PHONE)

Relationships:
VENDOR_REQUEST is sourced by **one** WAREHOUSE
WAREHOUSE can supply **many** VENDOR_REQUEST
WAREHOUSE has **many** SALE_PERSON
SALES_PERSON works at **one** WAREHOUSE

DEPP 420-56 ASSIGNMENT 1

Notes for #2 assumption on CUSTOMER (not essential reading)

Enforce one phone per customer ** see notes

1. CUS_FNAME, CUS_INITIAL, CUS_LNAME —> CUS_AREACODE, CUS_PHONE 1:1
2. Allow phone number identify more than one customer (spouse e.g.):
   CUS_AREACODE, CUS_PHONE does not —> CUS_FNAME, CUS_INITIAL, CUS_LNAME 1:*
3. Reason for #2 transitive:
   a:* possible for CUS_FNAME, CUS_INITIAL, CUS_LNAME: CUS_AREACODE, CUS_PHONE
   only if 1:* here
   CUS_FNAME, CUS_INITIAL, CUS_LNAME : CUS_CODE
   and
   CUS_FNAME, CUS_INITIAL, CUS_LNAME : CUS_CODE

____________which allows 1:* here
CUS_FNAME, CUS_INITIAL, CUS_LNAME : CUS_BALANCE

so don't allow 2 phones per person.
but can allow 2 persons per phone
which allows spouse to open second cust code under same home phone

A->B, C, D where
A = cust code
B = person
C = phone number
D = balance

B must —> C if we want to limit B to one A and limit B to one D
if B does not —> C then two C per B only possible when assign a second A to B
e.g. would be possible if B not —> C

A=1 B=3 C=4
A=2 B=3 C=5

but undesirable when A—>D because if B can have two C via two As then B can have two D via two As

e.g. would be possible if B not —> C

A=1 B=3 C=4 D=6
A=2 B=3 C=5 D=7

person must —> phone if want to limit person to one cust code and one balance

C not —> B ok if allow C to more than one A and allow C to more than one D
if C not—>B then a second B assigned to C allows requires second A assigned to C and a second D to B.

e.g. would be possible

A=1 B=3 C=5 D=6
A=2 B=4 C=5 D=7

phone can be assigned to more than one person
means a phone assigned to more than one account
means a phone to more than one balance
if e.g. a spouse opens a cust code under same phone number